Computer Science

LEVEL 3	15 TCE CREDIT POINTS
COURSE CODE	ITC315113
COURSE SPAN	2013 — 2017
READING AND WRITING STANDARD	NO
MATHEMATICS STANDARD	NO
COMPUTERS AND INTERNET STANDARD	YES

This course was delivered in 2017. Use A-Z Courses to find the current version (if available).

Computer Science involves the study of the storage, transformation and transfer of information

It includes both the theoretical study of algorithms and the practical problems involved in implementing them using the currently available technology. Computer Science can be considered a starting point for learners to continue further education and study in ICT or engineering as well as a preparation for learners in a vast range of careers that require efficient and effective use of ICT. Predicted ICT skills shortages, both within Australia and globally, point to the need for highly qualified professionals who have followed a computing career path and have skills far beyond ICT literacy. In addition, ICT is seen as a major driver of economic growth and productivity through its capacity to enhance efficiency and innovation.

Course Description

Through the study of Computer Science Level 3, learners will:

- gain a fundamental understanding of the software and hardware aspects of computing, and the underpinning mathematics and science of the discipline
- develop skills in the effective use of the tools used for the construction and documentation of software
- increase their general problem-solving capabilities and identify problems that are best solved by means of a computer
- develop an understanding and appreciation of the societal consequences of poor technological solutions, and the responsibilities of computing professionals
- develop an understanding of the discipline of computer science and apply this knowledge, and these skills to other disciplines
- develop skills in communicating effectively, to a range of stakeholders, a technological problem and its solutions
- increase their awareness of personal pathways in the area of Information Technology
- develop skills in managing one's own learning, including time management and organisation skills.

Learning Statement

Computer Science involves the study of the storage, transformation and transfer of information. It includes both the theoretical study of algorithms and the practical problems involved in implementing them using the currently available technology.

Computer Science can be considered a starting point for learners to continue further education and study in ICT or engineering as well as a preparation for learners in a vast range of careers that require efficient and effective use of ICT. Predicted ICT skills shortages, both within Australia and globally, point to the need for highly qualified professionals who have followed a computing career path and have skills far beyond ICT literacy. In addition, ICT is seen as a major driver of economic growth and productivity through its capacity to enhance efficiency and innovation.

Rationale

Learners will have an opportunity to explore future pathways related to this field of study and are provided with a window to possibilities within this area. Through the formal study of Computer Science, learners will gain an understanding of the discipline of computer science and its place in shaping the future of these technologies. To be prepared for a future characterised by change, students need to learn to solve problems creatively, manage and retrieve information, and communicate effectively. Computer Science not only gives learners the skills to utilise ICT, but begins to build information and technological problem solving skills. It enables learners to orient themselves towards the future with a sound understanding of ICT.

Computer Science is a rigorous field of study with unique characteristics. A key component of this course is algorithmic thinking. Learners are often exposed to numeric computation; however, they are often not given opportunities to develop formal step-wise algorithms for solving problems. Although algorithmic thinking is part of mathematics, not all algorithmic concepts are used in mathematics.

While there are a number of VET courses providing specific ICT skills and knowledge, the emphasis in this course is the underpinning knowledge and thinking skills which will provide a foundation for all learners in a rapidly changing field. The VET Certificates II & III in Information Technology do not include the explicit teaching of algorithm and program design.

Learning Outcomes

On successful completion of this course, learners will be able to:

1. gain an understanding of the application of Computer Science to a range of problems and appreciate the limitations of using algorithmic solutions
2. gain an understanding of the interaction between people and computers, and the implications for software design
3. design algorithmic, programming and technological solutions to a range of problems expressed in a variety of forms
4. develop skills in comparing, evaluating and refining algorithmic and programming solutions
5. gain an understanding of the software and hardware aspects of computing, and the underpinning mathematics and science of the discipline
6. develop skills in using a variety of technological resources such as online libraries and technical websites
7. gain an understanding of societal consequences of technological solutions and the professional responsibility of people working within this field
8. develop skills in providing effective communication to a range of stakeholders about technical problems and their solutions
9. develop skills in managing one's own learning, including time management and organisation skills
10. develop an awareness of career and further education opportunities.

Access

Learners need to be able to operate a computer in order to complete a range of computer programming exercises, access documentation and other relevant material on the internet.

Pathways

It is expected that learners participating in this course would have well developed ICT, numeracy and literacy skills. Experience in problem solving, including logical and critical thinking, would be advantageous.

It is envisaged that most learners wishing to pursue a computing career would use this as a starting point to study a degree at University, or VET Certificate IV, or Diploma, including combined Diploma/Degree courses. These courses may focus on multimedia and the internet, artificial intelligence, mobile and ubiquitous computing, systems and networks, computer security, distributed systems, software engineering or programming languages. Learners entering the workforce should expect to undergo further education and training.

Learners wishing to pursue careers in telecommunications or engineering will benefit from the problem solving skills and technical understanding developed in this course.

Learners may also choose to pursue a range of industry qualifications on completion of this course.

Resource Requirements

All learners will require access to computers with the following minimum requirements:

- access to the internet including web and email
- Java development kit and appropriate environment for creating and editing programs
- word processing facilities
- printing.

Course Size And Complexity

This course has a complexity level of 3.

At Level 3, the learner is expected to acquire a combination of theoretical and/or technical and factual knowledge and skills and use judgement when varying procedures to deal with unusual or unexpected aspects that may arise. Some skills in organising self and others are expected. Level 3 is a standard suitable to prepare learners for further study at tertiary level. VET competencies at this level are often those characteristic of an AQF Certificate III.

This course has a size value of 15.

Relationship To Other TASC Accredited And Recognised Senior Secondary Course

This course works well as a companion subject to Information Technology and Systems where students have the opportunity to learn wider project management skills and systems understanding. Other general computing courses at lower levels of complexity provide fundamental ICT skills which are pre-requisites for this course.

Fundamentals provided in electronics and the physical sciences provide learners with a deeper understanding of the function of computer components. Problem solving skills developed in mathematics complement those learned with Computer Science.

Computer Science is used to provide solutions to problems based in other disciplines.

Course Content

It is not envisaged that the following components be delivered as isolated topics but rather as an integrated body of knowledge. All listed content is compulsory.

PROBLEM SOLVING AND PROGRAMMING (70 HOURS)

Algorithms and programming solutions to a variety of problems are designed and expressed in a variety of forms. Learners will develop skills in understanding the problem, exploring problem solving strategies, design and creation of a solution. Algorithms that require mathematical solutions, such as those involving summation and searching, are investigated.

A fundamental understanding of the software development cycle (design, code, test, evaluate and refine) is required. Practical activities need to provide experience for learners in all stages of this cycle and to develop an understanding of the importance of analysis and design before beginning to code. Programs should adhere to established programming styles and be fully documented.

ALGORITHM DESIGN AND PROBLEM SOLVING

- Exploration of a range of problems - some problems cannot be reduced to an algorithm as they have no clear rules, some problems have clear rules but are difficult to put into algorithms because they have many possible responses to rules (e.g. Asian game of Go). Other problems are those for which rules are still to be discovered such as natural language processing and how to cure certain diseases. This section should provide a link between Computer Science and other disciplines
- Exploration of forms of problem solving – algebraic, algorithmic, trial and error
- The specific requirements of a problem are determined from problem definitions given in a variety of forms
- Examination of different solutions to the same problem (critical evaluation, most efficient, reliable)
- Problem deconstruction/decomposition – problems need to be broken down into a number of well defined steps
- Visualisation of solution – use of diagrams to illustrate the solution
- Expression of the solution to the problem – use an **initially/when** model for event-driven solutions.

PROGRAMMING

Students learn the fundamentals of Object Oriented Programming and event driven programming.

Learners write a variety of Java Applets using the following **Java** features:

- primitive types (promotion and casting)
- arithmetic and logic operators (+, -, *, / , %, &&, !, ||), order of operations, and some mathematical functions – such as (Math.pow(), Math.random())
- graphics (drawing, filling)
- pre-defined Objects, including arrays and Strings
- control-flow (selection using if else/switch, iteration using for/while)
- methods, parameters and scope
- GUIs (widgets including Buttons, TextField, Labels)
- events, and listeners using the AWT library
- classes and information hiding.

A key component of programming should be an emphasis on good programming practice. Programs need to adhere to a defined set of standards including good variable name choice, commenting, and indenting. Learners should be introduced to official requirements of particular organisations.

A key element of this topic is designing Applets for genuine solutions. Fundamental notions of HCI (Human Computer Interaction) are introduced.

TESTING AND EVALUATION

- A structured approach to testing is followed. Testing plans are written from program specifications in the absence of a program
- Tracing as a means of debugging programs is introduced, including both hand and automated tracing
- Self, peer and review by other adults is used to evaluate Applets and identify future refinements.

DOCUMENTATION

Programs need to be accompanied by both technical and user documentation. Technical documentation includes internal comments of programs. User documentation provides a description of the program’s purpose, operating instruction and appropriate online help.

COMPUTER FUNDAMENTALS AND COMPUTER LIMITATIONS (40 HOURS)

In order to come to an understanding of the limitations and possibilities for the use of computer technology into the future, learners need to understand computer architectures, and the role of the operating system.

Areas to be covered:

- binary number system for whole number and fraction and conversions to decimal and hexadecimal
- basic binary arithmetic (addition only)
- twos complement representation and arithmetic (addition and subtraction only)
- representation of primitive data types (integer, char, Boolean, float)
- representation of non-numeric data using hexadecimal where appropriate (e.g. characters, colours, instructions)
- implications of representation of floating point numbers for accuracy of calculations
- representation of arrays as well as sound and picture files
- Boolean operators (AND, OR, NOT)
- logic gates, basic computer circuits and the flip-flop
- using truth tables, Karnaugh maps and simplifications using the specified list of logic laws to design logic circuits
- computer architecture – the fundamental components of a computer in the von Neumann architecture
- machine code and its relationship to high level languages such as Java
- the machine cycle required to add two numbers (fetch, decode, execute)
- operating systems and the role of the JVM
- newer technologies and their relationship to basic computer architecture, such as multi core technology, and parallel computing.

SOCIAL/ETHICAL ISSUES AND PROFESSIONAL RESPONSIBILITY (10 HOURS)

Computer professionals have specialised knowledge and often have positions with authority. For this reason, they may have a significant impact on society, including many of the things that people value. Along with such power comes the duty to exercise that power responsibly.

Areas to be covered:

- career pathways, skills and education required
- the role of professional associations and codes of ethics
- responsibilities of the computing professional in the workplace
- responsibilities of those in positions of authority
- examples and consequences of technological errors, such as software bugs.

COMPUTING OPTION (30 HOURS)

The skills gained in computer science are used to explore an area of interest in more depth. The option chosen must enable learners to demonstrate problem solving skills, research, and technical communication skills. In addition, learners must adhere to ethical and professional standards as they are prescribed in the course. The option product will be used to assess criteria 8, 9 and 1 or 6, along with at least one other criterion.

Suggested topics:

- Production of a Java Applet for a client following the software development lifecycle
- Object oriented programming as applied in Flash ActionScripting
- Exploration of network programming
- Programming for mobile devices such as MIDlets
- Exploration of Java libraries (e.g. Greenfoot to build a simulation or interactive game, Robocode to build robots)
- Digital Electronics.

Assessment

Criterion-based assessment is a form of outcomes assessment that identifies the extent of learner achievement at an appropriate end-point of study. Although assessment – as part of the learning program – is continuous, much of it is formative, and is done to help learners identify what they need to do to attain the maximum benefit from their study of the course. Therefore, assessment for summative reporting to TASC will focus on what both teacher and learner understand to reflect end-point achievement.

The standard of achievement each learner attains on each criterion is recorded as a rating 'A', 'B', or 'C', according to the outcomes specified in the standards section of the course.

A 't' notation must be used where a learner demonstrates any achievement against a criterion less than the standard specified for the 'C' rating.

A 'z' notation is to be used where a learner provides no evidence of achievement at all.

Providers offering this course must participate in quality assurance processes specified by TASC to ensure provider validity and comparability of standards across all awards. To learn more, see TASC's quality assurance processes and assessment information.

Internal assessment of all criteria will be made by the provider. Providers will report the learner's rating for each criterion to TASC.

TASC will supervise the external assessment of designated criteria which will be indicated by an asterisk (*). The ratings obtained from the external assessments will be used in addition to internal ratings from the provider to determine the final award.

Quality Assurance Process

The following processes will be facilitated by TASC to ensure there is:

- a match between the standards for achievement specified in the course and the standards demonstrated by learners
- community confidence in the integrity and meaning of the qualifications.

Process – TASC gives course providers feedback about any systematic differences in the relationship of their internal and external assessments and, where appropriate, seeks further evidence through audit and requires corrective action in the future

External Assessment Requirements

The following criteria will be externally assessed: 1, 2, 3, 4 and 5.

For further information see the current external assessment specifications and guidelines for this course available in the Supporting Documents below.

Criteria

The assessment of Computer Science Level 3 will be based on the degree to which the learner can:

1. design and evaluate algorithmic solutions to a range of problems*
2. demonstrate knowledge of a high level programming language*
3. use appropriate objects in the design of programs*
4. demonstrate knowledge and understanding of computer architecture *
5. demonstrate knowledge and understanding of data representation and storage*
6. understand and apply the software development life cycle to a variety of problems
7. demonstrate understanding of the societal and professional responsibilities in the area of technology
8. plan, organise and complete activities
9. communicate technological information

* = denotes criteria that are both internally and externally assessed

Standards

Criterion 1: design and evaluate algorithmic solutions to a range of problems

This criterion is both internally and externally assessed.

Rating A	Rating B	Rating C
A learner applies the appropriate design theory to a given problem. A learner explores a range of possible solutions, evaluating each, and determines the best solution for the problem. The best solution is expressed in terms of an algorithm.	Given an unfamiliar computing task, a learner applies top down design in order to identify the required subtasks. A learner determines any missing events that are required by a solution. A learner explores a range of possible solutions and expresses a solution in algorithmic form.	Given an algorithmic solution to a problem, a learner determines whether it provides a correct solution to the problem for a given set of data. A learner modifies the algorithm to include additional features and writes algorithms to solve simple tasks.

Criterion 2: demonstrate knowledge of a high level programming language

This criterion is both internally and externally assessed.

Rating A	Rating B	Rating C
A learner demonstrates understanding of all required language constructs and interactions between components by writing and analysing Java code. This includes methods, parameters and scope of variables.	A learner makes informed and reasoned statements about the outcome of major features of the language(s). Appropriate strategies are used to determine the purpose of sections of code.	A learner makes informed statements about the outcome of commonly used features of the language(s). Small sections of a program are examined to determine their purpose. A learner traces sections of the programs using specified values.

Criterion 3: use appropriate objects in the design of programs

This criterion is both internally and externally assessed.

Rating A	Rating B	Rating C
A learner designs straightforward objects for a given situation. The objects can be modified to meet changed circumstances. Designed and modified objects are used within programming exercises.	A learner modifies given objects to meet changed circumstances, (e.g. arrays, strings). Given and modified objects are used within programming exercises.	A learner uses a variety of given objects within programming exercises.

Criterion 4: demonstrate knowledge and understanding of computer architecture

This criterion is both internally and externally assessed.

Rating A	Rating B	Rating C
A learner interrelates operating systems, machine architecture, fetch, code and execute cycle, machine code, binary, logical expressions and digital circuits. A learner gives extensive justification of limitations relating to computing tasks being performed.	A learner demonstrates understanding of information flow within a computer and relates hardware functions to higher order language constructs by converting between machine code and Java. A learner describes the information flow within a computer and relates hardware functions associated with the execution of machine code instructions. A learner devises a simple logic circuit utilising truth tables, Karnaugh maps and logical laws. The significance of the JVM is understood and related to the computer architecture. A learner describes and gives some justification of limitations relating to computing tasks being performed.	A learner describes the basic components of the computer and their function. Simple logic circuits are drawn from Boolean expressions and traced with values. The role of the Java Virtual Machine (JVM) is described. A learner describes the limitations imposed to computing tasks being performed.

Criterion 5: demonstrate knowledge and understanding of data representation and storage

This criterion is both internally and externally assessed.

Rating A	Rating B	Rating C
In addition to demonstrating the features of a 'B' rating, a learner demonstrates an understanding of data types including the storage issues associated with more complex data structures by describing and implementing calculation and storage situations where the limitations of the representation of data types and structures have an effect.	In addition to demonstrating the features of a 'C' rating, a learner demonstrates an understanding of the implications of the representation of all the primitive data types within a computer system and the limitations of each by describing and implementing calculation and storage situations where the limitations of the representation of primitive data types have an effect. A learner demonstrates an understanding of the special issues associated with floating point representation in the computer and how to deal with them in a program.	A learner describes the representation of the primitive data types in the computer. A learner demonstrates an understanding of the implications for the storage of integer data by describing and implementing calculation and storage situations where the limitations of the representation of integers has an effect. A learner is able to perform basic arithmetic in binary and Twos complement representation.

Criterion 6: understand and apply the software development life cycle to a variety of problems

Rating A	Rating B	Rating C
A learner writes well designed programs which meet the specifications using the specified standards with an appropriate user interface for the user. The most appropriate programming constructs are utilised. Relevant resources are identified, accessed and utilised. Comprehensive testing plans are specified before the program is written and programs are refined in response to the testing.	A learner follows the software development lifecycle to write programs that meet the specifications using the specified standards. Appropriate programming constructs are used and relevant resources accessed. Detailed testing plans are specified before the program is written and some program revisions are made after testing. Both hand and automated tracing are used to debug programs. Options for the user interface are explored with regard to the specified problem.	A learner writes straightforward programs which meet specifications using the specified standards and a range of programming constructs. Access is made to provided resources to assist in writing the program. Programs are tested against a plan and an evaluation is made on how well the program performs. Consideration is given to the user interface.

Criterion 7: demonstrate understanding of the societal and professional responsibilities in the area of technology

Rating A	Rating B	Rating C
Given a particular scenario, a learner identifies and justifies strategies to be put in place to reduce potential risks from security breaches, user errors and programming errors. A learner incorporates a range of strategies within their own solutions to reduce the potential for misuse (e.g. fully testing a program).	Given a particular scenario, a learner identifies the potential for misuse and relates this potential to professional organisations and codes of ethics. A learner is able to relate knowledge of program design and testing to minimise potential risks. A learner demonstrates an understanding of the impact of user interface design on users by discussing the impact of good and bad user interface design on users from a diverse range of abilities and backgrounds.	A learner describes a variety of technological pathways including the roles and the responsibilities of people working within these areas. A learner can recall relevant professional organisations and codes of ethics. A learner is aware of the concept of standards and the need to comply with standards. A learner gives examples of technological errors and the consequences of the errors across a range of areas.

Criterion 8: plan, organise and complete activities

Rating A	Rating B	Rating C
A learner determines and achieves goals by using appropriate strategies. A learner critically evaluates and adapts plans to ensure the outcomes can be achieved. Activities are undertaken in a constructive, collaborative and active manner. A learner designs instruments and constructively evaluates planning and processes.	A learner plans and organises activities to achieve given goals. A learner co-ordinates the tasks to complete activities within set times, and adapts plans to meet changed conditions. Activities are undertaken in a constructive, collaborative and active manner. Process and planning are critically evaluated.	A learner plans activities using provided resources to achieve the given goals and carry out the plans within a set time. A learner undertakes the activities in a collaborative and active manner, and contributes towards a critical evaluation of planning and processes.

Criterion 9: communicate technological information

Rating A	Rating B	Rating C
A learner conveys technological ideas and basic information in a variety of ways, (verbal, email, written documentation, online and screen information). A learner explores a variety of methods for conveying particular information, makes appropriate selections and devises techniques for testing the effectiveness of the communication. A learner devises methods for obtaining feedback. Revisions and alterations are made in response to feedback.	A learner conveys technological ideas and basic information in a variety of ways, (verbal, email, written documentation, online and screen information). A learner explores a variety of methods for conveying particular information and makes an appropriate selection. Communication is altered for particular audiences. Adjustments are made on the basis of feedback.	A learner conveys technological ideas and basic information in a variety of ways, (verbal, email, written documentation, online and screen information). This includes internal program documentation, screen design, technical and user documentation and problem solutions. A learners constructs and modifies communication to suit appropriate audiences.

Qualifications Available

Computer Science Level 3 (with the award of):

PRELIMINARY ACHIEVEMENT

SATISFACTORY ACHIEVEMENT

COMMENDABLE ACHIEVEMENT

HIGH ACHIEVEMENT

EXCEPTIONAL ACHIEVEMENT

Award Requirements

The final award will be determined by the Office of Tasmanian Assessment, Standards and Certification from 14 ratings (9 from the internal assessment, 5 from the external assessment).

The minimum requirements for an award in Computer Science Level 3 are as follows:

EXCEPTIONAL ACHIEVEMENT (EA)
12 'A', 2 'B' ratings (4 'A', 1 'B' from external assessment)

HIGH ACHIEVEMENT (HA)
6 'A', 6 'B', 2 'C' ratings (2 'A', 2 'B', 1 'C' from external assessment)

COMMENDABLE ACHIEVEMENT (CA)
8 'B', 5 'C' ratings (2 'B', 2'C' ratings from external assessment)

SATISFACTORY ACHIEVEMENT (SA)
12 'C' ratings (3 'C' from external assessment)

PRELIMINARY ACHIEVEMENT (PA)
6 'C' ratings

A learner who otherwise achieves the ratings for a CA (Commendable Achievement) or SA (Satisfactory Achievement) award but who fails to show any evidence of achievement in one or more criteria ('z' notation) will be issued with a PA (Preliminary Achievement) award.

Course Evaluation

The Department of Education's Curriculum Services will develop and regularly revise the curriculum. This evaluation will be informed by the experience of the course's implementation, delivery and assessment.

In addition, stakeholders may request Curriculum Services to review a particular aspect of an accredited course.

Requests for amendments to an accredited course will be forwarded by Curriculum Services to the Office of TASC for formal consideration.

Such requests for amendment will be considered in terms of the likely improvements to the outcomes for learners, possible consequences for delivery and assessment of the course, and alignment with Australian Curriculum materials.

A course is formally analysed prior to the expiry of its accreditation as part of the process to develop specifications to guide the development of any replacement course.

Course Developer

Department of Education, Tasmania. This course was developed in consultation with the University of Tasmania, Tasmanian Computer Science teachers, Polytechnic computing teachers and industry representatives.

Accreditation

The accreditation period for this course is from 1 January 2013 to 31 December 2017.

Version History

Version 1 – accredited on 14 November 2012. This course replaces Computer Science (ITC315108).

Supporting documents including external assessment material

- ITC315113 Asessment Report 2016.pdf (2017-07-21 01:05pm AEST)
- ITC315113 Assessment Report 2015.pdf (2017-07-21 01:05pm AEST)
- ITC315113 Exam Paper 2013.pdf (2017-07-21 01:05pm AEST)
- ITC315113 Exam Paper 2014.pdf (2017-07-21 01:05pm AEST)
- ITC315113 Exam Paper 2015.pdf (2017-07-21 01:05pm AEST)
- ITC315113 Exam Paper 2016.pdf (2017-07-21 01:05pm AEST)
- ITC315108 Assessment Report 2012.pdf (2017-07-26 02:21pm AEST)
- ITC315113 Assessment Report 2013.pdf (2017-07-26 02:22pm AEST)
- ITC315113 Assessment Report 2014.pdf (2017-07-26 02:22pm AEST)
- ITC315108 Exam Paper 2012.pdf (2017-07-26 02:23pm AEST)
- ITC315113 Examination Specifications 2013-2017.pdf (2017-07-26 02:25pm AEST)
- ITC315113 Info Booklet 2017.pdf (2017-09-11 11:50am AEST)
- ITC315113 Exam Paper 2017.pdf (2017-11-23 04:58pm AEDT)

Tasmanian Government